Weekly Report

2013-4-28

Feiran Wu

1 Introduction

Completed work:

1. Tested several Database framework and middleware;
2. Complete homework about ray tracing.

2 RESEARCH

Considering the limitation of the processing capability of the PC, we may need a series of component to deal with large scale data. One of the most important of them is database. So this week I have test a middleware LevelDB and a noSQL database CouchBase.

LevelDB is just a fast and lightweight key/value database library by Google. It own some features in noSQL DB, but quite simple. Strictly speaking, it is only a key/value query library with data compression and other function, yet far from a complete database. Chrome has integrate it for the IndexedDB implementation. In some case, when input large data in text format we may cost a lot of time parsing data. So if we parsing data to key/value format with LevelDB prior, it may boost data retrieval and reading.

Comparing to LevelDB, CouchBase is a totally noSQL memory database with JSON model. The deploying is quite simple and the CouchBase server may connect each other to be a cluster. Since it is a memory DB, the performance of CouchBase largely depends on the size of memory. In my test, I introduce two datasets. One is network data which has over 3 million records, and the other one is marine climate data in Gulf of Maine which totally has more than 100 thousand records.

After upload all datasets to server, the larger one occupies approximately 2GB and the other one occupies 200MB. 500MB memory is just assigned to database which means server has 500MB memory as data cache.

I randomly get 1 million records for each one and the result is quite different. In the smaller dataset, the performance is good since all data can be put in cache memory. However, the effect of the limitation of memory size is very obvious when I fetch data from the larger dataset. If I randomly get data in all 3 million records, the cache missed rate is always over 90%. Which means the server always swap data from memory and HDD. As a result, the 1 million fetch processing last even one hour. But if I just randomly fetch data with a narrow range, for instance, 1~10K, the processing will be slow in first queries and be faster with the increase number of queries (cache hit rate increase).

So we can conclude CouchBase as a development-friendly database and its performance is depended on the size of memory to some extents. In next step, I will use CouchBase as a data persistence framework to study multivariate data.

3 CONCLUSION

Next week's job :

Time-varying multivariate visualization. Datasets: Gulf of Maine, Global meteorological data from VAG weather group.